Spring 2020

Telecommunications Database

Tristan Hess
tlh173@zips.uakron.edu

Telecommunications Database

Tristan Hess

Williams Honors College, The University of Akron

Honors Research Project

Abstract:

The final goal of this project was to create a web application that is specifically tailored to make management of data for the Telecommunications department easier. The software that the department uses for phone service generates monthly reports that cannot be stored into their current database system. The Telecommunications Database project is a four-tier web application that was developed to store the monthly report information and alleviate the burden of manually searching through the reports for information. The web application implements basic database functionality for searching, inserting, updating, and deleting data contained in the monthly reports. Advanced searching functionality and a streamlined upload of all data in a report was developed as an alternative to importing data into a spreadsheet and using complicated search filters. This project was built on general knowledge of internet systems programming and database design, and can be further improved upon by being converted to modern high-level programming languages.

Keywords: Computer Science, Telecommunications, Database, Web Programming, MySQL, PHP

I. Introduction

The Telecommunications department at the University of Akron currently does not have a dedicated software to store and manage phone logs and reports. Reports are currently scattered throughout multiple spreadsheets across several destinations on the department's network. This makes managing current information, searching past information, and manually formatting the report data for use a massive headache for the Telecommunications staff. To help alleviate this issue, a single centralized database system must be in place that can serve as the main method of interacting with the stored data. It was crucial that each individual staff member could access the database and have multiple staff able to view it at the same time. For this type of functionality, a web application that connected to a database to manage and display the data was ideal. The web application would only need to be run on one computer on the network so that any other staff could access it as long as they knew the correct webpage to go to.

II. Implementation

A. Four-tier Web Application

This project was developed as a four-tier web application. These applications consist of four tiers: Web Client, Web Server, Application Server, and Database Server. The Web Client tier is what the user sees, it is the physical webpage that is loaded and interacted with. A Web Server is the browser that the webpage is hosted on, such as Google Chrome, Mozilla Firefox, or Microsoft Edge. The Application Server tier is hidden from the user and consists of three sub-tiers, typically described using the acronym MVC (Model-View-Controller). The MVC model makes up how a webpage is interpreted by the server. Different server-side languages for web

programming must be interpreted in different ways, and this layer will process everything to create a consistent webpage for the user to view and interact with. Finally, the Database Server tier contains all of the information needed to store data, retrieve data, and supporting software for other database features.

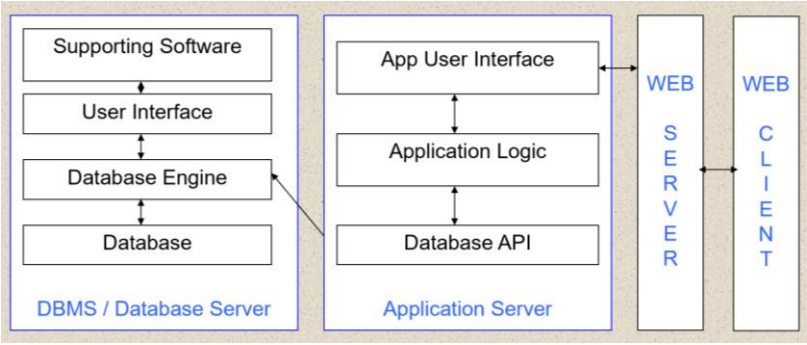

Figure 1- Four tier web application architecture

B. Front-end and Back-end

Front-end and Back-end are terms used in the Computer Science field to describe what a user sees and interacts with (front-end), and what goes on behind the scenes with server and database connections (back-end). The front-end was created with HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets), with some additional front-end JavaScript capabilities integrated into the application. HTML5 is the dominant programming language for creating webpages, and CSS produces the style and layout. JavaScript is a web programming language that has both front-end and back-end elements, but only the front-end ability to interact with the user's files was utilized. The back-end of the web application uses PHP (Personal Home Page) and a SQL (Structured Query Language) database. PHP is the programming language used to interact between the web browser and the database server tiers, and it requires a specific

interpreter to be used. A software called XAMPP is used for the Apache and MySQL modules, which interprets and runs the code for PHP and the SQL database.

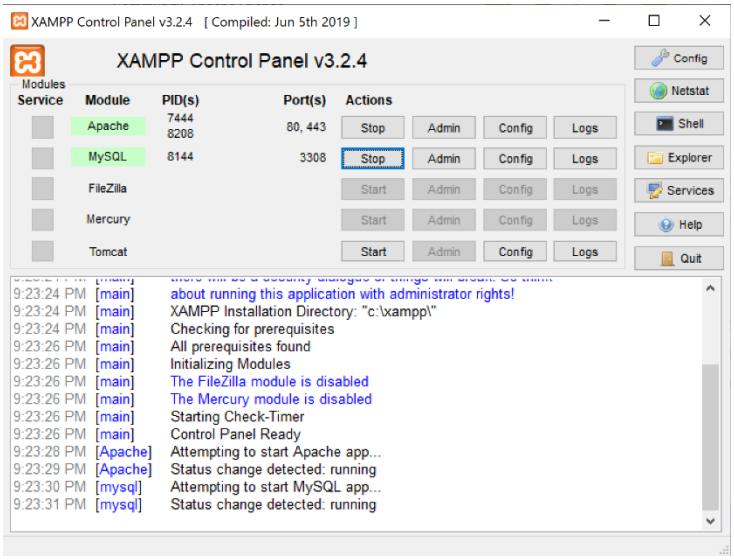

Figure 2- XAMPP control panel

C. Database Design

The database for the web application is designed in tables that correspond to the monthly reports that the telecommunications software generates. There are two different types of reports, an export_station report and a report_list report. The export_station reports contain information for the extension of the phone, its model, the name associated with it, etc. These types of reports are referenced for any sort of request that involves activating or deactivating extensions, or most commonly, moving extensions from one location to another. The second type of report, the report_list, contains the extension of the phone, permissions for admin features, restrictions on long distance calling, etc. A full breakdown of the database table design and report properties can be seen in the figure below.

Report_List	
extension : int(4)	> The 4 digit extension assigned to a phone
type : varchar(22)	> The classification of the phone (user, virtual, etc.)
cor : int(11)	> Permissions for long distance and international calling
tn : int(11)	> Permissions for conferencing
coverpath : varchar(3)	> Settings to redirect calls and ring pattern of the extension
name : varchar(30)	> The name associated with the extension
cos : int(11)	> Permissions for admin feature use (system admin, user, etc.)

Export_Station	
extension : int(4)	> The 4 digit extension assigned to a phone
type : varchar(7)	> The model of the phone
port : varchar(7)	> The port that the extension is assigned to on server boards
name : varchar(28)	> The name associated with the extension
room : varchar(5)	> The room the extension is assigned to
jack : varchar(5)	> The jack in the room the extension is assigned to
cable : varchar(5)	> The server's cable number associated with the extension
floor : varchar(7)	> The floor of the building the extension is assigned to
building : varchar(8)	> The building on campus the extension is assigned to

Figure 3- Database table design and properties

III. **Features**

A. Basic Functionality

The database can perform the basic functionalities of a database web application: insert, search, update, and delete. In order to perform one of the basic functions, the corresponding radio button must be selected on the page. This will show additional text boxes that correspond to the selected function that are used to carry out the database query. If the function should only be carried out on a specific database table, the radio button next to the table name must also be selected. The search function can be used to identify the extension of a specific phone if the other parameters are known. All other functions only need the extension to be given to function correctly. Below are the results for searching all tables for the name “Telecom” in the building “comp”.

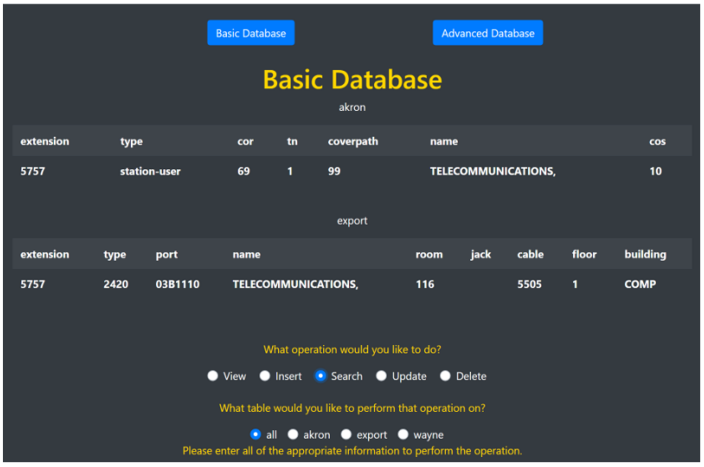

Figure 4- Example search function results

B. Advanced Functionality

The Advanced functionality page contains functions for more advanced database functions that would only see situational use. This page is used for finding unassigned extensions that match certain patterns, uploading information to the database, and finding the current lowest and highest extension assigned. When new extensions are assigned to employees, it is often desired that they conform to a certain pattern to match the rest of the extensions for that department. Assigning department extensions based on a specific pattern helps alleviate confusion when dealing with a large amount of changes to the system, such as the moving of a department from one location to another. Uploading information to the database is used when a new report is generated so that the database can automatically update all of its records with the selected report file. The search for the lowest and highest assigned extension is necessary because the university is only allotted a certain range of phone numbers it can use. This function is occasionally used to ensure that this range is not exceeded.

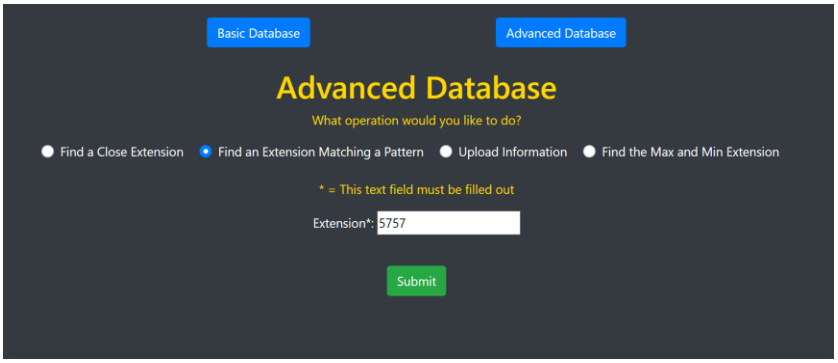

Figure 5- Advanced database functionality page

IV. Issues Encountered / Lessons Learned

The SQL database required a host to be able to run with a web application. A service called RemoteMySQL offered all of the functionality this project required for free. However, it was later discovered that the host would occasionally delete the database. Whenever this happened, I would have to create a new blank database and create the project's database tables all over again. I stored the SQL queries for creating the database tables in a separate file so that they could be easily run whenever this issue occurred.

```
CREATE TABLE `97EH6aKnCg`.`akron` (
  `extension` INT(4) NOT NULL,
  `type` VARCHAR(22) NOT NULL,
  `cor` INT NULL,
  `tn` INT NULL,
  `coverpath` VARCHAR(3) NULL,
  `name` VARCHAR(30) NULL,
  `cos` INT NULL,
  PRIMARY KEY (`extension`));

CREATE TABLE `97EH6aKnCg`.`wayne` (
  `extension` INT(5) NOT NULL,
  `type` VARCHAR(29) NOT NULL,
  `name` VARCHAR(30) NULL,
  `cor` INT NULL,
  `tn` INT NULL,
  `coverpath` VARCHAR(3) NULL,
  `cos` INT NULL,
  PRIMARY KEY (`extension`));

CREATE TABLE `97EH6aKnCg`.`export` (
  `extension` INT(4) NOT NULL,
  `type` VARCHAR(7) NOT NULL,
  `port` VARCHAR(7) NULL,
  `name` VARCHAR(28) NULL,
  `room` VARCHAR(10) NULL,
  `jack` VARCHAR(5) NULL,
  `cable` VARCHAR(5) NULL,
  `floor` VARCHAR(7) NULL,
  `building` VARCHAR(8) NULL,
  PRIMARY KEY (`extension`));
```

Figure 6- SQL queries for table recreation

The data in the reports could not be directly added into the database without parsing. Certain characters such as the single quote had to be removed in order for the data to be added to the database tables. SQL does not allow these characters because they are used in the syntax of queries, and would crash the application once one of these characters were read. Additionally, several fields of the reports were left blank by human error in the reports, so the data type for many of the parameters had to be a varchar(#) (Variable character string of length #). Previously, the cor and tn fields of the report_list reports were of the type int (integer). SQL does not allow blank entries for the int type, but it does allow blank entries for varchar. By changing the int types to varchar, the data was able to be directly inserted into the tables without the need of adding a placeholder or default value.

RemoteMySQL recently altered their database hosting services where the user must take a survey to create a database. The service cautions that this survey will automatically send and have access to personal information once you accept the terms and begin the survey. To counter this invasion of privacy, I was able to view the front-end website code using Google Chrome's built-in "Inspect Element" function by pressing F12, find the link that I would be redirected to once the survey was completed, and open it in a new browser tab. By doing this, I was able to create new databases whenever my old one was deleted without sharing personal information to ad companies.

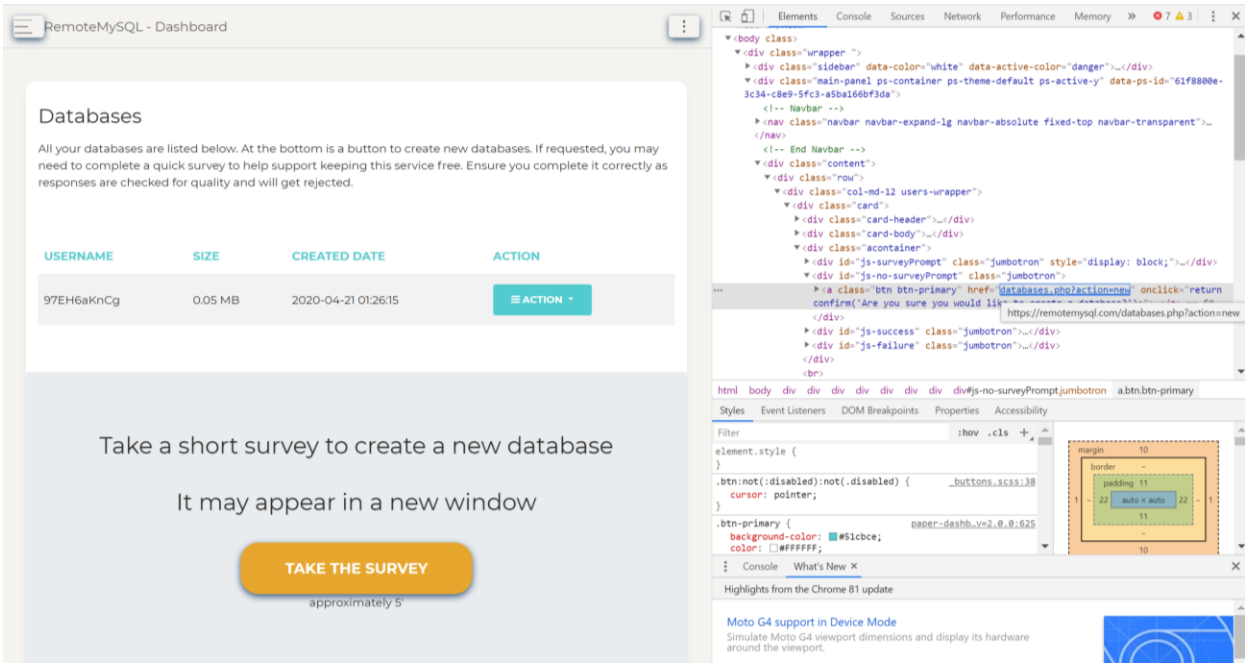

Figure 7- RemoteMySQL alternative database creation method

The database name and password must be included in the files for the web application, leaving them exposed for anyone who has access to the files to see. This is a major security concern if this application were to be commercialized.

PHP is not able to interact with the filesystem on a user's computer. In order to have the database directly read and communicate with the raw data files, a technology called AJAX (Asynchronous JavaScript and XML) had to be used. This allowed the front-end capabilities of JavaScript to work with the back-end capabilities of PHP and SQL.

V. Future Research

SQL injection is a type of attack on web application databases which compromises the security of a text input box to gain access to the internals of the database. These attacks can be used to view private database information, make unauthorized changes to data, or delete a database in its entirety. The web application currently uses a procedural style of connection that

is vulnerable to SQL injection. There is an object-oriented style of connection that is not vulnerable to SQL injection, but is significantly more complex to implement. It is ideal to convert the connection of the web application to the object-oriented style in order to prevent accidental or deliberate SQL injection attacks.

The database currently requires the installation of the XAMPP control panel to run on the host computer. This installation is currently blocked due to security implementations on the department computers. The database should be migrated to a new host so that the host computer is not required to do any additional software installations. Another reason why the database should be migrated is because of the occasional deletion of RemoteMySQL databases under the free user accounts. The database should be migrated to a database hosting service that does not run locally and does not delete its databases after inactivity.

This web application and database is structured in accordance to the four-tier web application architecture. The application currently uses a combination of HTML, JavaScript, PHP, AJAX, and SQL. These 5 different programming languages all utilize different syntax and different interpreters in order to compile and run properly. The same and more advanced functionality can be achieved with just two higher level languages: C# and ASP.NET. These two technologies differ in power from the older ones because they are able to create dynamic web pages, rather than static ones. Dynamic web pages are desired because they can implement a fast, intuitive, and visually appealing web application with much less code. The downside to these high-level languages is that they require a substantial time investment to learn and become proficient in.

VI. Conclusion

This project served as a way for me to combine my work experience with the Telecommunications department with the skills and knowledge I developed throughout my major. After working part-time for three years, I feel that I was able to give something back to the university and the Telecommunications staff that helped me along my journey. I expanded my knowledge of web applications and databases to a practical real-world scenario, and was able to study and implement new technologies I have never used. I developed a project that is able to alleviate some of the strain on the Telecommunications staff, as well as showcase my web development skills to potential employers.

References

Garcia-Molina, H., Widom, J., & Ullman, J. D. (2009). *Database systems: the complete book* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Sebesta, R. W. (2015). *Programming the World Wide Web* (8th ed.). Boston: Pearson.

Yingcai, X. (2018) *Introduction to Database* [PowerPoint Slides].